

Syntax Diagram Guide

Appendix C. Syntax Diagram Guide

Syntax Diagrams

Syntax diagrams are read from left to right. A single path is followed until the end mark is encountered. An end mark indicates nothing more can be entered with that command.

Within syntax diagrams, the following conventions are used:

- Diagram items that must be entered literally on the command line are in **bold**. These items include the command name, flags, and literal characters.
- Variable diagram items for which you determine the name to enter are in *italics*. These items include parameters that follow flags and parameters that the command reads, such as *files* and *directories*.
- Default values that do not have to be entered are in the normal font on a **bold** path.

Command Only

The simplest syntax diagram illustrates a command that is entered on the command line with nothing else. For example:

logname —|

OL805145

The bold command name means **logname** should be entered literally. As you follow the line away from the command name to the right, you reach the end mark and must stop. To enter this command, you would enter:

logname

Commands with Required Parameters

Many diagrams have parameters to represent specific values that you must enter on the command line. When you encounter a parameter, read the command description to determine what to enter in place of the parameter. The following diagram requires a parameter:

unlink — *file* —|

OL805227


The bold command name means **unlink** should be entered literally. As you follow the line away from the command name to the right, the next item that you encounter is the parameter *file*. Since the only path goes through this parameter, you must supply a file name. As you move further to the right, you reach the end mark and must stop. To enter this command, enter:

unlink *report*

Suppose you want to unlink two files, **report** and **memo**. Since this command accepts one and only one file, you would have to enter the command twice:

```
unlink report
unlink memo
```

Some commands allow you to enter more than one parameter on the command line. If you are allowed to do so, the diagrams show it with a **repeat arrow**, an arrow that provides a path back to an earlier part of the diagram. For example:

```
sact 
```

OL805063

The bold command name indicates **sact** should be entered literally. As you follow the line away from the command name to the right, the next item that you encounter is the parameter *file*. Since the only path goes through this parameter, you must enter a file name. As you move further to the right, you reach a repeat arrow. Here you can choose to remain on the main path and proceed to the end mark or follow the repeat arrow around to the point between the command name and the parameter. Following the repeat arrow allows you to select the parameter again. If there is a maximum number of parameters that you can enter, the diagram tells you that number. If no maximum is specified (as in this diagram), then you can choose the repeat arrow again and again until you reach the limit of the length of a command line. Here are some examples:

```
sact s.letter
sact s.letter s.memo s.report
```

Commands with an Optional Flag or Parameter

Many commands have optional flags or parameters. If something is optional, you have a choice of paths in the diagram. One takes you around the item, and the other takes you through it. For example:

```
del 
```

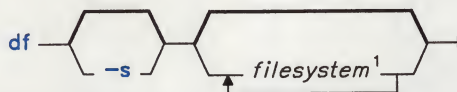
OL805049

With this diagram, as you move to the right from the command name, you reach a branch. You can either take the upper branch (nothing is entered) or you can take the lower branch (enter the flag). The bold line above the flag is the default line. This is the most commonly used path. After the branch, you encounter a parameter. Since the only path goes through the parameter you must enter it. After the parameter is entered once you can choose to proceed to the end mark or use the repeat arrow to select the parameter again.

For example:

```
del file1
del file1 file2 file3
del - file1 file2
```

As the command syntax becomes more complicated, the features of the diagrams are combined to help you enter commands properly. The next diagram shows a command that accepts an optional flag and an optional parameter that can be repeated.



¹ The default action is to provide information for each file system in `/etc/filesystems` with the attribute `free=true`.

OL805052

In this diagram, as you move to the right from the command name, you reach a branch. You can either take the upper branch (nothing is entered) or the lower branch (enter the flag). Next you encounter another branch. Here you can either take the upper branch (nothing is entered) or the lower branch (enter the parameter). If you choose to enter the parameter, you can enter the parameter once and proceed to the end mark or you can use the repeat arrow to select the parameter again. For example:

```
df
df -s
df system1
df -s system1
df system1 system2
df -s system1 system2 system3
```

Commands That Take More Than One Flag

With many commands, you can enter as many items from a group of flags or parameters as you want within the limits of the length of the command line. If this is the case, the items are in a box that has a repeat arrow around it. Follow the arrow around and through the box as many times as necessary to select all of the items you want to use. Note that most commands do not work properly if you choose the same flags more than once. Therefore, once you have chosen an item from the box, you should not choose it again unless a footnote indicates a flag may be used more than once.

For example:



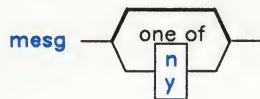
OL805086

With this command you can enter only the command name by following the default line over the box, enter one flag and then continue to the end mark, or follow the arrow around and choose both flags. The following are examples of valid command lines:

```
cat
cat -u
cat -s
cat -u -s
```

Commands with an Exclusive Flag or Parameter

Many commands have flags or parameters that should not be entered together on the command line. Mutually exclusive items are enclosed in a single-choice box (a box with the words one of above it). You can choose only one item from this type of box. The following diagram contains a single-choice box.



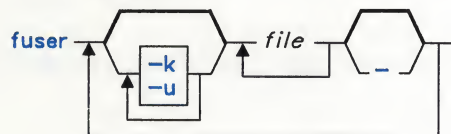
OL805036

Valid ways to enter this command are:

```
mesg
mesg n
mesg y
```

Commands that Can Repeat Part of a Sequence

Some commands allow you to choose flags for each parameter that they read. When this is the case, more than one repeat arrow allows you to go back to earlier parts of the diagram. For example:



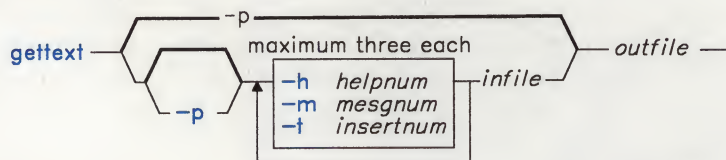
In this diagram, there are three repeat arrows. The first allows you to choose one or both flags. The second allows you to have **fuser** read more than one file. The third allows you to repeat the complete sequence from the beginning of the diagram to the end. The following are all correct ways to enter **fuser**:

```
fuser memo
fuser memo -
fuser -k memo
fuser -k -u memo
fuser -k -u memo letter
fuser -k -u memo -
fuser -k memo - -u -k letter -u report -
```

The third arrow allows you to enter the same flag repeatedly, but only *after* at least one file name has been entered. If you follow the diagram, you cannot repeat a flag without entering at least one file name after it.

Commands with Default Values

The default line can show more than just an alternate path around flags and parameters. Sometimes, a flag is set automatically (by default) or a sometimes a parameter has a default value. When this is the case, the default value is shown in the normal font on the default line. For example:



OL805130

If you do not enter any flags with **gettext**, the **-p** flag is set by default. If you choose the path that contains the **-h**, **-m**, and **-t** flags, you must choose whether or not to use the **-p** flag also. The following command lines are equivalent ways of entering **gettext**:

```
gettext -p report
gettext report
```

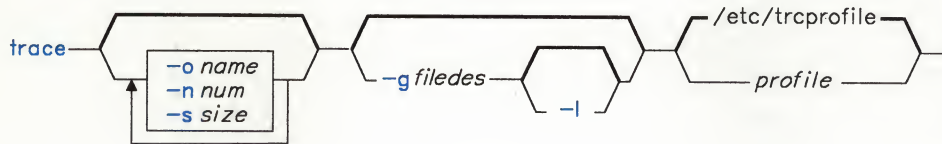
The following are valid command lines that do not use the default flag for **gettext**:

```
gettext -h2 report
gettext -m3 memo report
```

To select both the **-p** and **-m** flag, you must explicitly enter the **-p** flag. For example:

```
gettext -p -m3 memo
```

You can also have default parameter values.



OL805279

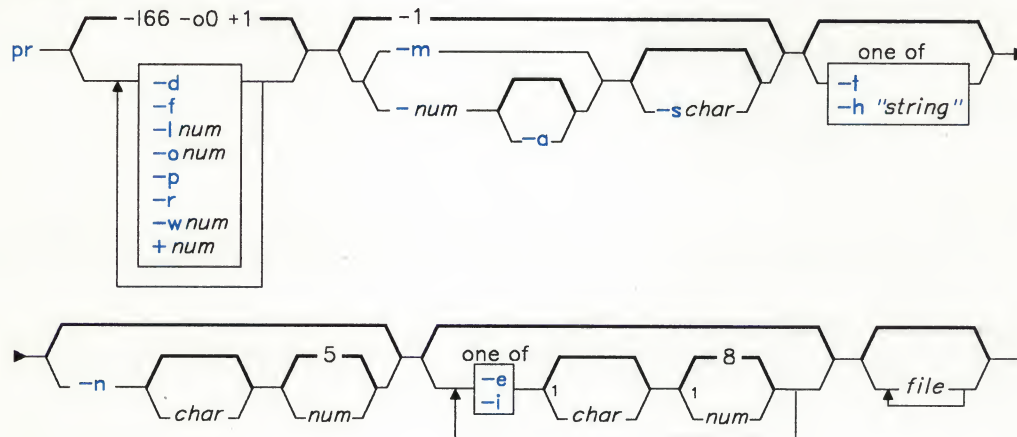
In this case you can choose to specify a *profile* or to use the default value */etc/trcprofile*. Since the bold path indicates that this is a default, you do not have to enter this file name. The system does this for you. If a *profile* is not supplied, *trace* reads the file */etc/trcprofile*.

The following are equivalent command lines:

```
trace
trace /etc/trcprofile
```

Diagrams That Are Continued on the Next Line

Some of the more complex diagrams cannot fit on one line. They are marked with an arrowhead where they break, and they continue with the arrowhead on the next line. For example:



¹ Do not put a blank between these items.

OL805437

Follow the diagram making choices until you reach the arrowhead. Then go down to the arrowhead on the next line. Continue until you reach the end mark. Following the diagram will seem to impose a specific order to the flags; however, you do not need to strictly follow that order when entering the command. If strict order is important, it is stated under "Description" in the commands discussion.

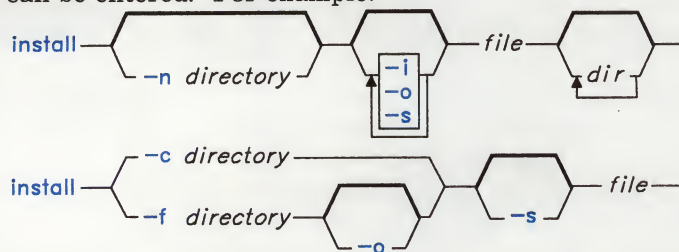
The following are some of the ways you can enter **pr** on the command line:

```
pr
pr -d
pr -o4 -r -m -sX memo letter
pr -r -m -t -n4 -iX3 memo letter report
pr -m -n4 -r -iX3 -t memo report letter
pr -l30 5 -3 -a -nX -iX3 -eY memo report
```

Notice that this diagram has a footnote. Footnotes are used to show information that cannot be diagramed. In this case, it tells you that you cannot put a space between the **-e** or **-i** flags and their parameters.

Commands With More Than One Diagram

Some commands require more than one diagram to indicate the different ways a command can be entered. For example:



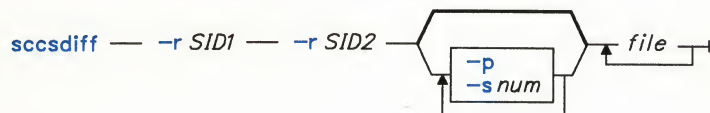
OL805022

Because some flags and parameters cannot be combined with others, two diagrams are required to indicate the ways the command can be entered. For example, the following are ways you can enter this command:

```
install -o fixit /etc /games
install -n /usr/bin fixit
install -c /usr/bin fixit
install -f /usr/bin -o -s fixit
```

File Input and Output in Diagrams

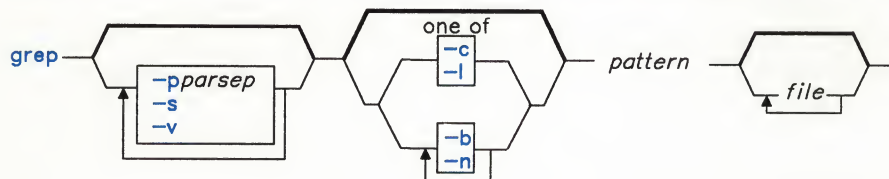
Some commands must read a file as their input, some must read standard input, and some can read both. The syntax diagrams help you to determine which case applies to each particular command. If a command must read a file as its input, the diagram shows a path through a parameter representing the file and the “Description” section tells you this file is an input file. The path in the diagram will not have a branch around it. The following is an example of a command that must read an input file:



OL805258

When there is no place in a diagram to supply an input file, the command reads standard input for this information. To supply input through standard input, you can supply it through a pipeline, through redirection of the output from another command, or directly from the keyboard (if it is standard input).

Most commands can either read standard input or files for their input. The diagrams show this by branching and giving you a choice of entering a file name or nothing. For example:



OL805375

If you specify an input file, the command reads that file for its input. If you do not specify a file, the command reads standard input. The following are valid command lines for this command:

```
grep AAA
grep -sAAA
grep AAA memo
grep -s -v AAA memo
grep AAA memo letter report
```

In the first two cases, the **grep** command reads standard input. In the last three cases, the **grep** command uses the specified input files.

Note: Some commands require that you enter a - (minus) when you want the command to read standard input. If this is the case, it is explained under “Description”, not in the diagram. Usually this is done so that you can read several files as input and you can include standard input as one of the files.

Appendix D. Japanese Language Support

The following table lists the commands that have not been modified to support Japanese characters.

acct/*	dist	ipcrm
acctcms	dos	ipcs
acctcom	dosdel	ipctable
acctcon1, acctcon2	dp	istat
acctdisk, acctdusg	dsipc	keyboard
acctmerg	dsldxprof	lint
acctprc1, acctprc2, accton	dsstate	locator
acctman	dsxlate	m4
adb	dumpfmt	mail, Mail
ali	edconfig	mailstats
anno	edit	makekey
ap	eqn, neqn, checkeq	man
arithmetic	errdead	mark
as	errdemon	mdrc
back	errpt, errpd	mhl
bellmail	errstop	mhmail
bffcreate	errupdate	mhpath
bj	ex	mm, checkmm
burst	factor	mmt, mant, mmt, mvt, checkmm
cc, fcc, vcc	fish	moo
cd	fnt	msgchk
cflow	folder	msh
chkcomp	folders	ncheck
chngstate	fortune	ndtable
col	forw	newgrp
comp	fptype	next
confer, joinconf	fuser	nroff, troff
conflict	fwtmp, acctwtmp, wtmpfix	number
connect	gdev - hpd, erase, hardcopy, tekset, td	open
cpp	ged	packf
craps	gend	pick
ct	gettext	post
ctags	graphics	prev
cu	greek	prof
cvid	groups	profiler - prfld, prfstat, prfdc, prfsnap, prfpr
Cvt	gutil - bel, cvrtopt, gd, gtop, pd, ptog, quit, remcom, whatis, yoo	prompter
cw, checkcw	hangman	proto
cxref	hp	pstart, pshare, pdelay
dbx	hyphen	ptx
deroff	inc	puttext
diffmk	install	pwttable
diskusg	install-mh	quiz
display	installp	rc

rcvdist
rcvpack
rcvstore
rcvtty
refile
repl
rmail
rmf
rmm
runacct
sact
scan
send
sendmail, mailq, newaliases
setdma
show
slocal
sno
sortm
sound
spell, hashmake, spellin,
hashcheck
splp
spost

stat
syslogd
tapechk
tbl
tc
termdef
test
tic
tlog
tlogger
toc - dtoc, ttoc, vtoc
tplot
trace
trcrpt
trcstop
trcupdate
trdiag
ttt
turnon, turnoff
ugtable
umask
units
updatep, inudocm, inuupdt
uuchek

uucico
uucleanup
uulog
uuname
uupick
uusched
uustat
uuto
uutry, Uutry, uukick
uux
uuxqt
varyoff
varyon
verify
vmh
vrml2rtfont
vrmlconfig
whatnow
whom
writesrv
wump
xdbx
300, 300s
4014
450

